ATTORNEY'S DOCKET NO.: S01022.81054

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Pierre-Yvan LIARDET, Fabrice ROMAIN, Yannick TEGLIA and Laurence SIRTORI
Serial No.:	10/611,254
Filed:	July 1, 2003
For:	CYPHERING/DECYPHERING PERFORMED BY AN INTEGRATED CIRCUIT
Examiner:	Unassigned
Art Unit:	Unassigned

Confirmation No. Unassigned

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

Transmitted herewith for filing is/are the following document(s):

[X] Certified Copy of French Priority Application No. 02 08268
[X] Return Post Card

If the enclosed papers are considered incomplete, the Mail Room and/or the Application Branch is respectfully requested to contact the undersigned collect at (617)720-3500, Boston, Massachusetts.

No check is enclosed. If it is determined that a fee is necessary, the fee may be charged to the account of the undersigned, Deposit Account No. 23/2825. A duplicate of this sheet is enclosed.

CERTIFICATE OF MAILING UNDER 37 C.F.R. §1.8(a)

I hereby certify that this document is being placed in the United States mail with first-class postage attached, addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on July 24, 2003.

Gail Driscoll

Respectfully submitted,

Pierre-Yvan Liardet et al., Applicants

By:

James H. Morris
Reg. No.: 34,681
WOLF, GREENFIELD & SACKS, P.C.
600 Atlantic Avenue
Boston, Massachusetts 02210
Tel. (617) 720-3500

Attorney Docket No.: S01022.81054
XNDD

INPI
INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE

BREVET D'INVENTION

CERTIFICAT D'UTILITÉ - CERTIFICAT D'ADDITION

COPIE OFFICIELLE

Le Directeur général de l'Institut national de la propriété industrielle certifie que le document ci-annexé est la copie certifiée conforme d'une demande de titre de propriété industrielle déposée à l'Institut.

Fait à Paris, le 10 JUIN 2003

Pour le Directeur général de l'Institut
national de la propriété industrielle
Le Chef du Département des brevets

Martine PLANCHE

INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE
SIEGE
26 bis, rue de Saint Petersbourg
75800 PARIS cedex 08
Téléphone : 33 (0)1 53 04 53 04
Télécopie : 33 (0)1 53 04 45 23
www.inpi.fr

INPI
INSTITUT NATIONAL DE LA PROPRIÉTÉ INDUSTRIELLE

26 bis, rue de Saint Pétersbourg
75800 Paris Cedex 08
Téléphone : 01 53 04 53 04 Télécopie : 01 42 94 86 54

BREVET D'INVENTION
CERTIFICAT D'UTILITÉ

Code de la propriété intellectuelle-Livre VI

cerfa N° 55 -1328

REQUÊTE EN DÉLIVRANCE 1/2

Réservé à L'INPI

Cet imprimé est à remplir lisiblement à l'encre noire

REMISE DES PIÈCES
DATE 2 JUIL 2002
LIEU 38 INPI GRENOBLE
N° D'ENREGISTREMENT NATIONAL ATTRIBUÉ PAR L'INPI 0208268
DATE DE DÉPÔT ATTRIBUÉE PAR L'INPI - 2 JUIL 2002

Vos références pour ce dossier
(facultatif) B5532

❶ NOM ET ADRESSE DU DEMANDEUR OU DU MANDATAIRE À QUI LA CORRESPONDANCE DOIT ÊTRE ADRESSÉE

Cabinet Michel de Beaumont
1 rue Champollion
38000 GRENOBLE

Confirmation d'un dépôt par télécopie ☐ N° attribué par l'INPI à la télécopie

❷ NATURE DE LA DEMANDE	Cochez l'une des 4 cases suivantes
Demande de Brevet	[X]
Demande de certificat d'utilité	☐
Demande divisionnaire	☐
Demande de brevet initiale	N° Date / /
ou demande de certificat d'utilité initiale	N° Date / /
Transformation d'une demande de brevet européen *Demande de brevet intiale*	☐ N° Date / /

❸ **TITRE DE L'INVENTION (200 caractères ou espaces maximum)**

CHIFFREMENT/DÉCHIFFREMENT EXÉCUTÉ PAR UN CIRCUIT INTÉGRÉ

❹ **DÉCLARATION DE PRIORITÉ OU REQUÊTE DU BÉNÉFICE DE LA DATE DE DÉPÔT D'UNE DEMANDE ANTÉRIEURE FRANÇAISE**	Pays ou organisation Date N° Pays ou organisation Date / / N° Pays ou organisation Date / / N° ☐ **S'il y a d'autres priorités, cochez la case et utilisez l'imprimé "Suite"**
❺ **DEMANDEUR**	☐ **S'il y a d'autres demandeurs, cochez la case et utilisez l'imprimé "Suite"**
Nom ou dénomination sociale	STMicroelectronics SA
Prénoms	
Forme juridique	Société anonyme
N° SIREN	
Code APE-NAF	
ADRESSE Rue	29, Boulevard Romain Rolland
Code postal et ville	92120 MONTROUGE
Pays	FRANCE
Nationalité	Française
N° de téléphone *(facultatif)*	
N° de télécopie *(facultatif)*	
Adresse électronique *(facultatif)*	

INPI
INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE

BREVET D'INVENTION
CERTIFICAT D'UTILITÉ
Code de la propriété intellectuelle-Livre VI

REQUÊTE EN DÉLIVRANCE 2/2

Réservé à L'INPI

REMISE DES PIÈCES
DATE 2 JUIL 2002
LIEU 38 INPI GRENOBLE
N° D'ENREGISTREMENT 0208268
NATIONAL ATTRIBUÉ PAR L'INPI

Vos références pour ce dossier : *(facultatif)* B5532		
⑥ MANDATAIRE		
Nom		
Prénom		
Cabinet ou Société		Cabinet Michel de Beaumont
N° de pouvoir permanent et/ou de lien contractuel		
ADRESSE	Rue	1 Rue Champollion
	Code postal et ville	38000 GRENOBLE
N° de téléphone *(facultatif)*		04.76.51.84.51
N° de télécopie *(facultatif)*		04.76.44.62.54
Adresse électronique *(facultatif)*		cab.beaumont@wanadoo.fr
⑦ INVENTEUR (S)		
Les inventeurs sont les demandeurs		☐ Oui ☒ Non **Dans ce cas fournir une désignation d'inventeur (s) séparée**
⑧ RAPPORT DE RECHERCHE		**Uniquement pour une demande de brevet (y compris division et transformation)**
Établissement immédiat ou établissement différé		☒ / ☐
Paiement échelonné de la redevance		**Paiement en trois versements, uniquement pour les personnes physiques** ☐ Oui ☒ Non
⑨ RÉDUCTION DU TAUX DES REDEVANCES		**Uniquement pour les personnes physiques** ☐ Requise pour la première fois pour cette invention *(joindre un avis de non-imposition)* ☐ Requise antérieurement à ce dépôt *(joindre une copie de la décision d'admission pour cette invention ou indiquer sa référence) :*
Si vous avez utilisé l'imprimé "Suite", indiquez le nombre de pages jointes		

⑩ SIGNATURE DU DEMANDEUR OU DU MANDATAIRE
(Nom et qualité du signataire)

Michel de Beaumont
Mandataire n° 92-1016

VISA DE LA PREFECTURE OU DE L'INPI

D.R.GR

La loi n°78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés s'applique aux réponses faites à ce formulaire. Elle garantit un droit d'accès et de rectification pour les données vous concernant auprès de l'INPI.

CHIFFREMENT/DÉCHIFFREMENT EXÉCUTÉ PAR UN CIRCUIT INTÉGRÉ

La présente invention concerne le domaine du chiffrement de données numériques au moyen d'algorithmes destinés à masquer des données d'origine afin de les rendre indétectables par un pirate éventuel. L'invention concerne plus particulièrement les algorithmes mettant en oeuvre une même transformation sur différentes parties des données à coder.

Les algorithmes de chiffrement/déchiffrement auxquels s'applique la présente invention sont généralement exécutés par des circuits intégrés, soit au moyen de machines d'états en logique câblée, soit au moyen de microprocesseurs exécutant un programme en mémoire (généralement une mémoire morte). De tels algorithmes utilisent des clés secrètes propres aux circuits intégrés ou à l'utilisateur, qui sont exploitées par l'algorithme pour coder les données.

Un exemple d'algorithme de chiffrement/déchiffrement auquel s'applique tout particulièrement la présente invention est un algorithme connu sous la dénomination AES (Advanced Encryption Standard, FIPS PUB 197). Cet algorithme applique à un mot ou code de données découpé en blocs, une même transformation plusieurs fois de suite à partir de clés de chiffrement différentes ou, plus précisément, de parties d'un mot binaire constituant une clé.

La figure 1 illustre, par un organigramme simplifié, les étapes principales d'un algorithme classique de type AES. On se contentera de décrire le chiffrement, le déchiffrement reprenant les transformations inverses.

Cet algorithme chiffre un mot ou code S_0 d'un nombre de bits prédéterminé (généralement, 128 bits) en un autre mot ou code S_n de même taille. Les données à chiffrer sont en fait constituées de plusieurs mots ou codes résultant d'un découpage préalable des données en mots ayant tous la même taille. Le chiffrement et le déchiffrement reposent sur une clé secrète dont la longueur (généralement de 128 à 256 bits) conditionne la sécurité du chiffrement.

En pratique, chaque étape d'un algorithme de type AES traite une matrice de quatre lignes et quatre colonnes, représentant un mot et dont chaque élément est un octet ou bloc du code de 128 bits traité. Pour simplifier la description qui va suivre, on fera référence, pour chaque étape, à un état considéré comme étant une matrice.

Pour la mise en oeuvre de l'algorithme de chiffrement ou de déchiffrement, on commence par produire, à partir de la clé secrète sur 128 bits, 11 sous-clés comprenant chacune également 128 bits. De façon plus générale, à partir d'une clé secrète d'un nombre m de bits, on dérive n+1 sous-clé K0, ... Ki, ... Kn de m bits chacune. Ces sous-clés sont destinées à être utilisées par l'algorithme comme cela sera décrit ci-après en relation avec la figure 1.

On part d'un état initial (bloc 1, STATE INIT) S_0 du code ou mot de données à chiffrer.

Une première phase du procédé de chiffrement est une opération (bloc 2, ADDROUNDKEY) dite de "blanchiment" qui consiste à effectuer une combinaison de type OU-Exclusif (XOR) de l'état initial S_0 avec la première sous-clé K0. On obtient un premier état intermédiaire S_1.

Une deuxième phase du procédé de chiffrement consiste à effectuer plusieurs tours ou cycles d'une même transformation

T faisant intervenir, à chaque tour, l'état S_{i-1} obtenu au tour précédent et une sous-clé courante Ki. Le nombre de tours de la transformation T correspond à n-1, c'est-à-dire au nombre de sous-clés dérivées, diminué de 2.

Chaque transformation de tour T est constituée de quatre opérations appliquées successivement. La figure 2 illustre plus en détail ces quatre opérations sur une matrice 20 de quatre lignes et quatre colonnes d'octets binaires à laquelle s'applique un algorithme de type AES.

Une première étape (bloc 3, SHIFTROWS) consiste à opérer une rotation sur les trois dernières lignes de la matrice 20. La première ligne 201 de la matrice 20 demeure inchangée. La deuxième ligne 202 subit une rotation d'un octet. La troisième ligne 203 subit une rotation de deux octets. La quatrième ligne 204 subit une rotation de trois octets.

Une deuxième étape (bloc 4, SUBBYTES) de la transformation de tour T constitue une transformation non linéaire dans laquelle chaque octet de la matrice 20' constituant l'état courant est remplacé par son image prise dans une table de substitution (SBOX). Comme l'illustre la figure 2, la table de substitution SBOX est obtenue par deux transformations successives. Une première transformation (bloc 41, INV) consiste à inverser l'octet considéré (l'élément de la matrice 20') dans le corps fini d'ordre 2^8 (pour correspondre à l'octet), l'octet 00 constituant sa propre image. Cette inversion est suivie d'une transformation affine (bloc 42, AFFINE).

Des exemples de transformation non-linéaire de substitution telle que celle exposée ci-dessus sont décrits, par exemple dans l'ouvrage "The Design of Rijndael" de Joan Daemen et Vincent Rijmen, paru aux éditions Springer-Verlag (ISBN 3-540-42580-2) et dans la norme AES (FIPS PUB 197).

La troisième étape (bloc 5, MIXCOLUMNS) de la transformation de tour T consiste à considérer chaque colonne de la matrice 20" issue de l'étape précédente comme un polynôme sur le

corps fini d'ordre 2^8, et à multiplier chacun de ces polynômes par un polynôme de combinaison P[X] modulo un polynôme M[X].

La quatrième et dernière étape de la transformation de tour T de rang i consiste à appliquer la sous-clé Ki à la matrice résultante 20" de l'état précédent pour obtenir une matrice 20"', dans laquelle chaque élément de la matrice 20" a été combiné par un OU-Exclusif, bit à bit, avec la sous-clé Ki (bloc 6, ADDROUNDKEY). Cette étape 6 est la même que l'étape 2 de la première phase du chiffrement, mais effectuée avec une sous-clé différente.

A l'issu de l'étape 6, on obtient, pour un tour de rang i, un état $S_i=T(Ki, S_{i-1})$. Les quatre étapes de la transformation de tour sont répétées n-1 fois, c'est-à-dire qu'après l'étape 6, on revient à l'étape 3 pour ré-effectuer un tour avec une clé suivante.

La troisième phase de l'algorithme de chiffrement (figure 1) consiste en quelque sorte en un dernier tour, légèrement modifié par rapport à celui illustré par la figure 2. En fait, on reproduit les étapes de la transformation de tour à l'exception de la troisième (MIXCOLUMNS). Cela revient à effectuer successivement des étapes 7, 8 et 9 correspondant aux étapes 3, 4 et 6 décrites précédemment avec, comme clé pour l'étape 9, la dernière sous-clé Kn.

On obtient alors l'état $S_n=T'(Kn, S_{n-1})$. Ce résultat est finalement mis en forme (bloc 10, RESULTFORM) pour utilisation ultérieure.

Une faiblesse connue des implémentations sur carte à puce des algorithmes de type AES, ou plus généralement des algorithmes mettant en oeuvre plusieurs tours ou cycles d'une même transformation (T) sur un code découpé en blocs, est la sensibilité aux attaques par analyse de la consommation en courant du circuit exécutant l'algorithme. Une telle attaque connue sous la dénomination DPA (Differential Power Analysis) consiste à corréler la consommation du circuit intégré exécutant l'algorithme avec les clés secrètes utilisées lors du chiffrement ou du déchif-

frement. En pratique, à partir d'un message à chiffrer et d'hypothèses sur la clé secrète, on établit une courbe de corrélation statistique en fonction du temps entre la consommation du produit pour le chiffrement du message et une valeur intermédiaire calculée par le circuit. De telles attaques en consommation sont décrites dans la littérature (voir par exemple, l'article "Differential Power Analysis" de Paul Kocher, Joshua Jaffe et Benjamin Jun, paru en 1999, Conférence CRYPTO 99, pages 388-397, publié par Springer-Verlag LNCS 1666).

Une solution connue, pour rendre les algorithmes plus résistants contre des attaques par analyse de la consommation du circuit intégré, consiste à faire intervenir un nombre aléatoire dans l'exécution de l'algorithme. Le recours à une valeur aléatoire consiste à masquer l'état au début de l'algorithme par cette valeur aléatoire et à rétablir le résultat escompté à la fin de l'algorithme.

La figure 3 illustre, de façon partielle et très schématique, une première technique connue d'introduction d'un nombre aléatoire Rd dans l'exécution d'un algorithme de type AES. Partant d'un état initial de la matrice (bloc 11, STATEINIT), on effectue une combinaison de type OU-Exclusif bit à bit (bloc 12, +) par un nombre aléatoire Rd. Ce nombre est donc introduit avant l'étape 2 de combinaison avec la première sous clé K0. On doit ensuite tenir compte de ce nombre aléatoire Rd à certains stades de l'algorithme. Tout d'abord, lors des étapes 4 et 8 de transformation non linéaire (SUBBYTES), on doit utiliser une table de substitution ($SBOX_{Rd}$) tenant compte du nombre aléatoire. Puis, à chaque transformation de tour, après l'introduction de la clé courante Ki (étape 6), on doit effectuer une combinaison (bloc 13, +) de type OU-Exclusif avec le nombre Rd. De plus, après l'étape 13, on combine (bloc 15, +) par un OU-Exclusif le résultat obtenu avec une grandeur MC(SR(Rd)) correspondant à l'application des fonctions SR de décalage de lignes (SHIFTROWS) et MC de mélange de colonne (MIXCOLUMNS) au nombre Rd.

Après la dernière transformation T', la combinaison (bloc 16, +) par un OU-Exclusif du résultat obtenu avec la valeur SR(Rd) correspondant à l'application du décalage de lignes à la valeur Rd permet de retrouver le résultat escompté.

Le recours obligatoire à une table de substitution fonction du nombre aléatoire oblige de recalculer cette table à chaque chiffrement ou déchiffrement. Ce recalcul des tables de substitution, nécessaire pour obtenir une bonne résistance aux attaques DPA, entraîne un besoin de mémoire important dans le circuit intégré et allonge le temps d'exécution de l'algorithme par le temps de calcul nécessaire. Par exemple, pour des codes (matrices) sur 128 bits, le recalcul d'une table de substitution $SBOX_{Rd}$ pour chaque octet de l'état requiert 16 tables de 256 octets, ce qui représente 4 kilo-octets de mémoire. Une telle mémoire est loin d'être négligeable lorsqu'elle est intégrée, par exemple, dans une carte à puce.

La figure 4 illustre une deuxième solution classique pour faire intervenir une valeur aléatoire dans un algorithme de chiffrement de type AES. Cette solution est décrite dans l'article "An implementation of DES and AES, secure against some attacks" De M.L. Akkar et C. Giraud publié à la conférence CHES 2001 (éditions Springer-Verlag).

Cette solution consiste à remplacer le recours à des tables de substitution par des transformations calculées à chaque tour de l'algorithme. Le résultat est le même, en ce sens qu'il conduit à une substitution des différents octets de la matrice. Ce qui change, c'est la manière d'obtenir cette substitution.

Selon cette solution, on utilise deux nombres aléatoires Rd1 et Rd2 que l'on fait intervenir à différentes étapes de l'algorithme. Le premier nombre aléatoire Rd1 intervient au départ (entre les blocs 1 et 2) et est additionné (combinaison de type OU-Exclusif 22). La deuxième valeur aléatoire Rd2 est introduite dans les transformations de tour qu'il s'agisse des n-1 transformations identiques T ou de la dernière transformation T'.

Le résultat issu de l'étape 3 ou 7 de décalage de ligne est combiné par une multiplication polynomiale 23 à coefficients sur le corps fini d'ordre 2^8 (modulo un polynôme irréductible) avec la valeur aléatoire Rd2. Puis, la matrice résultante obtenue est additionnée (combinaison de type OU-Exclusif) à une matrice représentant le résultat de l'opération précédente (S_i*Rd2). Cette addition est symbolisée par un bloc 25 en figure 4.

Les deux opérations précédentes sont effectuées avant l'étape 24 de substitution d'octets qui comprend ici essentiellement deux transformations. Une première transformation (bloc 241, INV) consiste à inverser chaque octet de la matrice résultante de l'étape 25. Puis, on additionne (OU-Exclusif) à cette matrice inverse (bloc 242, +), le produit (octet par octet modulo un polynôme irréductible) de l'état initial S_i par l'inverse ($Rd2^{-1}$) de la valeur aléatoire. Le résultat est ensuite multiplié (bloc 243, X) par la valeur aléatoire Rd2. Là encore, il s'agit d'une multiplication polynomiale. Enfin, la dernière étape de la substitution d'octets 24 de la matrice consiste en une transformation affine 244 (AFFINE). En sortie de l'étape 24, la matrice résultante est soumise à l'étape d'addition de la sous-clé correspondante (étape 6 ou 9).

Si on est dans une transformation de tour, l'étape suivant l'étape 24 est l'étape 5 (MIXCOLUMNS). Puis, après l'étape 6, on combine (bloc 26, +) par un OU-Exclusif le résultat obtenu avec la valeur Rd1. Le résultat de l'addition 26 est combiné (bloc 27), toujours par un OU-Exclusif, avec le résultat (MC(AF(SR(Rd1)))) du traitement polynomial de mélange de colonnes (MC) de la transformation affine AF appliquée au décalage de ligne SR appliqué à la valeur Rd1.

Si l'on est dans la dernière transformation T', l'étape suivant l'étape 24 est l'étape 9 avec la clé Kn. Enfin, on combine (bloc 29, +) par un OU-Exclusif le résultat obtenu avec le résultat (AF(SR(Rd1))) de la transformation affine AF appliquée au décalage de ligne SR appliqué à la valeur Rd1. La

sortie du bloc 29 fournit l'état devant être mis en forme par l'étape 10.

Une telle solution requiert moins de mémoire que la première solution classique illustrée en relation avec la figure 3. Toutefois, elle augmente considérablement le temps d'exécution de l'algorithme. En effet, à chaque tour de l'algorithme, l'opération correspondant à la substitution devient complexe et requiert de nombreuses opérations modulo un polynôme.

Le problème de traitement par un nombre aléatoire vient essentiellement du fait que, dans un algorithme du type auquel l'invention s'applique, l'opération de substitution est une opération non linéaire.

La présente invention vise à proposer une nouvelle solution à l'introduction d'au moins une grandeur aléatoire dans un algorithme de chiffrement de type AES qui pallie les inconvénients des solutions connues. Plus généralement, l'invention vise à proposer l'introduction d'au moins une valeur aléatoire dans un algorithme faisant subir à un code ou mot d'entrée, découpé en blocs, plusieurs fois la même transformation (par matrice de substitution) avec des clés différentes.

L'invention vise également à proposer une solution qui minimise le nombre de fois qu'une table de substitution doit être calculée et/ou mémorisée.

L'invention vise également à minimiser le temps de calcul nécessaire à l'exécution de l'algorithme suite à l'introduction du nombre aléatoire.

Pour atteindre ces objets et d'autres, la présente invention prévoit un procédé de chiffrement et/ou déchiffrement, par un circuit intégré, d'un code numérique d'entrée au moyen d'au moins une première clé, consistant :

à découper ledit code en plusieurs blocs de données de mêmes dimensions ; et

à appliquer auxdits blocs au moins un tour de chiffrement ou déchiffrement consistant à faire subir à chaque bloc

au moins une même transformation non linéaire et à combiner ultérieurement chaque bloc avec ladite clé,

les opérandes étant masqués, lors de l'exécution du procédé, au moyen d'au moins un premier nombre aléatoire (R1) ayant la taille dudit code et dont tous les blocs ont la même valeur en combinant, par une fonction de type OU-Exclusif, les blocs d'entrée et de sortie de la transformation non linéaire avec ledit nombre aléatoire.

Selon un mode de mise en oeuvre de la présente invention, plusieurs tours de chiffrement sont appliqués avec une clé différente à chaque tour.

Selon un mode de mise en oeuvre de la présente invention, le code d'entrée est combiné avec un deuxième nombre aléatoire de même dimension que ce code.

Selon un mode de mise en oeuvre de la présente invention, ladite transformation non linéaire consiste à utiliser une table de substitution des blocs du code d'entrée, calculée avec un troisième nombre aléatoire de même longueur que ledit code et dont tous les blocs ont la même valeur. Selon l'invention, ladite table respecte le fait que la transformation d'un code d'entrée, préalablement combiné par un OU-Exclusif avec le premier nombre aléatoire, correspond au résultat de la combinaison par un OU-Exclusif de ce code d'entrée avec ledit troisième nombre aléatoire.

Selon un mode de mise en oeuvre de la présente invention, le procédé est appliqué à un algorithme de chiffrement de type AES.

Selon un mode de mise en oeuvre de la présente invention, ledit premier nombre aléatoire est changé à chaque tour de chiffrement.

Selon un mode de mise en oeuvre de la présente invention, ledit deuxième nombre aléatoire est changé à chaque chiffrement d'une nouvelle donnée.

Selon un mode de mise en oeuvre de la présente invention, ledit troisième nombre aléatoire est changé à chaque tour de chiffrement.

L'invention prévoit également un circuit intégré comprenant un bloc de chiffrement/déchiffrement d'une donnée d'entrée découpée en blocs de mêmes dimensions, comportant :

des moyens de génération d'au moins un premier nombre aléatoire de même taille que la taille des blocs de la donnée d'entrée ; et

des moyens de combinaison dudit nombre aléatoire avec chaque bloc, en entrée et en sortie d'une transformation non linéaire mise en oeuvre par le chiffrement/déchiffrement.

Ces objets, caractéristiques et avantages, ainsi que d'autres de la présente invention seront exposés en détail dans la description suivante de modes de mise en oeuvre de réalisation particuliers faite à titre non-limitatif en relation avec les figures jointes parmi lesquelles :

la figure 1 décrite précédemment illustre par un organigramme schématique, un procédé de chiffrement classique du type auquel s'applique la présente invention ;

la figure 2 décrite précédemment illustre les traitements opérés sur un état matriciel dans une transformation de tour du procédé de la figure 1 ;

la figure 3 décrite précédemment représente les étapes d'un premier procédé classique de prise en compte d'un nombre aléatoire dans un algorithme de chiffrement du type de celui illustré par la figure 1 ;

la figure 4 décrite précédemment représente une deuxième solution classique d'introduction de nombres aléatoires dans un algorithme de chiffrement du type de celui représenté en figure 1 ;

la figure 5 illustre, par un organigramme schématique, un mode de réalisation d'un algorithme de chiffrement selon la présente invention ; et

la figure 6 représente, par un organigramme schématique, un mode de mise en oeuvre d'un procédé de déchiffrement selon la présente invention.

Pour des raisons de clarté, seules les étapes qui sont nécessaires à la compréhension de l'invention ont été représentées aux figures et seront décrites par la suite. En particulier, les traitements en amont et en aval de l'algorithme de chiffrement n'ont pas été détaillés et ne font pas l'objet de l'invention. De plus, les opérations de subdivision de la quantité secrète en plusieurs sous-clés à prendre en compte par l'algorithme, ainsi que la génération des nombres aléatoires adaptés n'ont pas été détaillées et sont à la portée de l'homme du métier à partir des indications qui seront données ci-après.

Selon l'invention, on utilise une grandeur aléatoire ayant la même taille que l'état à chiffrer (la matrice) pour les transformations traitant plusieurs octets en même temps ou qui les mélangent entre eux, comme c'est le cas pour les transformations de type mélange de colonnes, introduction de sous-clé et décalage de lignes.

Par contre, on n'utilise pas cette grandeur aléatoire pour les fonctions non linéaires, telles que celle mise en oeuvre pour la substitution d'octets par une table de substitution dans le cas considéré. Selon l'invention, on masque la table de substitution par une autre valeur aléatoire dont les octets (ou plus généralement les blocs d'une taille correspondant à la taille des blocs du code pris en compte dans la table de substitution) sont tous identiques. Bien qu'elle soit donc effectuée qu'au moyen d'une grandeur aléatoire d'un octet, une telle opération de masquage est efficace dans la mesure où, grâce au masquage complet des fonctions opérant sur la totalité du bloc, il n'est pas possible pour un pirate d'exploiter cette particularité au travers d'une fonction de corrélation. Selon une variante de réalisation, on utilise également une valeur pseudoaléatoire, liée à cette valeur aléatoire, pour masquer la table de substitution.

La figure 5 représente un organigramme d'un mode de mise en oeuvre d'un algorithme de type AES, masqué au moyen de valeurs aléatoires et/ou pseudoaléatoires selon la présente invention.

Dans la description qui suit, on fera référence aux tailles de mots binaires en prenant l'exemple d'un algorithme AES utilisant une clé de 128 bits et un découpage d'un code d'entrée de 128 bits sous la forme d'une matrice de quatre lignes et de quatre colonnes d'octets. On notera cependant que tout ce qui sera décrit par la suite s'applique quel que soit la taille des clés et des codes d'entrée et de sortie, pourvu que les relations éventuelles entre ceux-ci soient respectées. En particulier, on notera que la taille des valeurs aléatoires (le cas échéant pseudoaléatoires) utilisées pour la table de substitution doit correspondre à la taille d'un élément de la matrice, alors que la taille de la valeur aléatoire (le cas échéant pseudoaléatoire) utilisée pour les transformations linéaires doit correspondre à la taille d'un état d'entrée complet. Par définition, on désigne par l'addition, une combinaison de type OU-Exclusif et par la multiplication, une multiplication polynomiale modulo un polynôme irréductible.

En partie gauche de la figure 5 ont été représentées les étapes successives de l'algorithme de chiffrement tandis qu'en partie droite de cette figure ont été indiqués les états obtenus à l'issue de chaque étape.

On part d'un état initial (bloc 31, STATE INIT). Cet état S_0 correspond au code (données) à chiffrer par l'algorithme.

La première étape 41 consiste à effectuer une combinaison de type OU-Exclusif de l'état S_0 avec une valeur aléatoire R dont la taille est la même que l'état S_0 (par exemple, 128 bits).

Puis, on exécute une étape classique 32 d'addition de sous-clé (bloc 32, ADDROUNDKEY) par une combinaison de type

OU-Exclusif de la première sous-clé K0 avec le résultat de l'étape précédente. L'état obtenu correspond à l'état S_1+R.

On entre alors dans la deuxième phase du procédé de chiffrement consistant à exécuter n-1 tours d'une même transformation T. Cette transformation fait intervenir les étapes du procédé classique (dans cet exemple, l'AES) que l'on souhaite masquer par au moins une valeur aléatoire. Dans l'exemple représenté, il s'agit des étapes successives 33 de décalage de lignes (SHIFTROWS), 34 de substitution d'octets (SUBBYTES) au moyen d'une table de substitution SBOX, 35 de mélange de colonnes (MIXCOMUMNS) et 36 de combinaison de type OU-Exclusif (ADDROUNDKEY) avec la sous-clé Ki de rang i.

Selon l'invention, entre ces étapes, on introduit deux valeurs aléatoires R1 et R2 (le cas échéant, R1=R2) constituées chacune de suites d'octets de même valeur. Le nombre (par exemple, 16) d'octets de chaque valeur correspond au nombre d'octets d'un état traité (par exemple, 128 bits).

En sortie de l'étape 33, on obtient une matrice ayant des lignes décalées à partir de la matrice d'état S_i combinée à la grandeur aléatoire R. En désignant par SR la fonction de décalage de lignes, on peut écrire que l'on obtient à l'issu de l'étape 33 : $SR(S_i+R)=SR(S_i)+SR(R)$.

Selon l'invention, avant d'effectuer la substitution de l'étape 34, on combine (bloc 42) l'état $SR(S_i)$+SR avec une valeur de même taille (R1+SR(R)) correspondant à l'application du décalage de ligne à la valeur aléatoire R (SR(R)) combinée, octet par octet, par un OU-Exclusif avec la valeur aléatoire R1. En d'autres termes, l'état est masqué par une valeur de même taille dont chaque octet a la même valeur aléatoire.

On exécute alors l'étape 34 de substitution octet par octet au moyen de la table de substitution $SBOX_{R1,R2}$. Cette table est, selon l'invention, fonction de la valeur R2 et est liée à la valeur R1 en respectant la relation suivante :

$SB(S_i+R1)=SBOX(S_i)+R2$, où SBOX représente la table de substitution de l'algorithme que l'on souhaite masquer et SB

désigne la fonction de substitution d'octets (SUBBYTES). En d'autres termes, on calcule une nouvelle table de substitution SB à partir de la table SBOX de l'algorithme que l'on souhaite masquer par les valeurs R1 et R2.

Au résultat $(SB(SR(S_i))+R2)$ de l'étape 34 qui correspond à un état masqué par la valeur R2 (chaque octet de la matrice est masqué par un octet de même valeur), on applique une combinaison de type OU-Exclusif (bloc 43) avec la combinaison OU-Exclusif R2+R (octet par octet) de la valeur sur 128 bits R et de l'octet R2.

Le résultat $(SB(SR(S_i))+R)$ subit la transformation 35 de mélange de colonnes de l'algorithme classique. Toujours en respectant l'algorithme classique, la sous-clé Ki est introduite par l'étape 36 de combinaison de type OU-Exclusif avec la matrice précédente. Le résultat $MC(SB(SR(S_i)))+MC(R)+Ki$, où MC désigne la fonction de mélange de colonnes MIXCOLUMNS, est combiné (bloc 44) avec une matrice correspondant à la somme (combinaison de type OU-Exclusif) de la grandeur aléatoire R et de cette même grandeur MC(R) ayant subi une transformation de mélange de colonnes identique à la transformation 35.

La transformation cyclique se termine par cette étape 44 à l'issue de laquelle, selon le rang i, on revient en étape 33 pour une nouvelle itération ou l'on passe à l'étape 37 de décalage de lignes (SHIFTROWS) de la dernière transformation T'.

Là encore, l'invention consiste à intercaler, entre certaine étapes de l'algorithme dont on souhaite masquer l'exécution par des valeurs aléatoires, des combinaisons logiques des matrices traitées par les grandeurs R1 et R2.

La transformation par matrice de substitution 38 est identique à celle décrite en relation avec l'étape 34, mais encadrée par des combinaisons 45 et 46. Ces combinaisons sont identiques aux combinaisons 42 et 43 décrites précédemment, en amont et en aval de la transformation 34.

A l'issue de l'étape 46, la matrice obtenue $SB(SR(S_{n-1}))+R$ est combinée avec la dernière sous-clé Kn (bloc 39). Puis, on

rétablit le résultat escompté ($SB(SR(S_{n-1}))+Kn=T'(S_{n-1}, Kn)$ en recombinant (bloc 47, +) par un OU-Exclusif la matrice obtenue par la première grandeur aléatoire R de même taille que cette matrice. On met alors en forme le résultat S_n de façon classique (bloc 40, RESULTFORM).

Un avantage de la présente invention est que les quantités R1 et R2 ainsi que la table de substitution SBOX peuvent être recalculées à chaque tour T de la transformation cyclique ou à chaque chiffrement ou déchiffrement de la donnée d'entrée par l'algorithme complet.

Un autre avantage de l'invention est qu'une mémoire correspondant au double de la matrice à traiter est suffisante pour stocker les nouvelles tables de substitution (l'ancienne et la nouvelle) dans la mesure où celles-ci sont combinées avec une valeur aléatoire dont la taille correspond à celle d'un élément de la matrice.

La figure 6 représente, sous forme d'organigramme simplifié, un mode de mise en oeuvre d'un algorithme de déchiffrement d'une donnée S_n selon l'invention.

Comme dans l'algorithme dont on souhaite masquer l'exécution par les valeurs aléatoires, le déchiffrement reprend les étapes inverses à celles du chiffrement à l'exception des étapes d'introduction des clés ou sous-clés Ki, qui s'effectue dans l'ordre inverse.

L'état initial (bloc 51, STATE INIT) correspond ici à un état chiffré ou crypté (S_n) des données.

Comme pour l'algorithme de chiffrement, on commence par combiner (bloc 61) l'état initial avec une quantité aléatoire R ayant la même taille que la donnée initiale. Puis, on combine (bloc 52, ADDROUNDKEY) l'état obtenu S_n+R avec la sous-clé Kn qui correspond à la dernière partie de la clé de chiffrement (dans cet exemple, le dernier octet). L'état obtenu $S_{n-1}+R$ est alors soumis à n-1 cycles d'une même transformation de déchiffrement prenant en compte à chaque tour une sous-clé Ki de rang inférieur.

Les étapes successives se déduisent des étapes de chiffrement décrites précédemment :

- décalage de lignes inverse (bloc 53, INVSHIFTROWS, fonction ISR) correspondant à la transformation inverse de celle du bloc 33 ;
- combinaison de type OU-Exclusif 62 avec la quantité aléatoire R1 et la fonction 53 appliquée à la quantité aléatoire R (ISR(R)) ;
- application (bloc 54, INVSUBYTES, fonction ISB) de la table de substitution $INVSBOX_{R1,R2}$ inverse à celle utilisée lors de l'étape 34 ;
- combinaison 63 de type OU-Exclusif avec les quantités R2 et R ;
- transformation inverse (bloc 55, INVMIXCOLUMNS, fonction IMC) de la transformation dite de mélange de colonnes 35 ;
- combinaison de type OU-Exclusif (bloc 56, ADDROUNDKEY) avec la sous-clé Ki de rang i ; et
- combinaison 64 de type OU-Exclusif avec la quantité aléatoire R et avec le résultat d'une transformation 55 (IMC) appliquée à cette quantité aléatoire.

Les étapes 62, 63 et 64 sont identiques aux étapes 42, 43 et 44 exécutées lors du chiffrement. Les étapes 52, 53, 54, 55 et 56 correspondent aux fonctions mises en oeuvre classiquement pour le déchiffrement de l'algorithme dont on souhaite masquer l'exécution.

A l'issue du dernier tour de cette transformation cyclique, on applique successivement des étapes 57, 58 et 59, inverses aux étapes de chiffrement 37, 38 et 39 et correspondant aux étapes classiques du procédé de déchiffrement, en intercalant les mêmes étapes de combinaison 65, 66 et 67 que lors du chiffrement.

On obtient alors la matrice résultat S_0 correspondant aux données déchiffrées.

Les quantités aléatoires (ou pseudoaléatoires) R, R1, et R2 utilisées lors du déchiffrement des données peuvent être différentes de celles utilisées lors du chiffrement. De plus, comme lors de l'exécution de l'algorithme de chiffrement, ces quantités aléatoires peuvent être différentes à chaque tour de la transformation cyclique.

Selon un mode de mise en oeuvre préféré, les quantités R1 et R2 sont identiques.

Selon une variante de réalisation, l'opération du bloc 43 (figure 5) est remplacée par un Ou-Exclusif avec la quantité R2+IMC(R). L'étape 44 est alors supprimée. De façon similaire, en figure 6, on peut supprimer l'étape 64 en modifiant l'étape 63 en introduisant R2+MC(R) au lieu de R2+R.

Bien entendu, la présente invention est susceptible de diverses variantes et modifications qui apparaîtront à l'homme de l'art. En particulier, l'invention qui a été décrite ci-dessus en relation avec l'algorithme de chiffrement de type AES pourra être transposée à tout algorithme de chiffrement dont le code d'entrée est découpé en blocs de tailles identiques pour être chiffré, chaque bloc subissant une même transformation non linéaire.

De plus, l'adaptation de l'invention et des tailles des quantités aléatoires aux tailles des données traitées et des clés utilisées est à la portée de l'homme du métier. On veillera à respecter un nombre de sous-clés correspondant aux nombres de tours et une taille de quantité aléatoire R correspondant à la taille des sous-clés, donc des blocs. Par ailleurs, les nombres indiqués comme aléatoires peuvent être issus d'un générateur pseudoaléatoire.

Enfin, l'invention s'applique quelle que soit l'utilisation faite des données chiffrées.

Un exemple particulier d'application de l'invention concerne la mise en oeuvre d'un algorithme de chiffrement/déchiffrement de type AES dans une carte à puce.

REVENDICATIONS

1. Procédé de chiffrement et/ou déchiffrement, par un circuit intégré, d'un code numérique d'entrée (S_0, S_n) au moyen d'au moins une première clé (Ki), consistant :

à découper ledit code en plusieurs blocs de données de mêmes dimensions ; et

à appliquer auxdits blocs au moins un tour (T) de chiffrement ou déchiffrement consistant à faire subir à chaque bloc au moins une même transformation non linéaire (SUBBYTES, INVSUBBYTES) et à combiner ultérieurement chaque bloc avec ladite clé (Ki),

caractérisé en ce qu'il consiste à masquer les opérandes lors de l'exécution du procédé au moyen d'au moins un premier nombre aléatoire (R1) ayant la taille dudit code et dont tous les blocs ont la même valeur en combinant, par une fonction de type OU-Exclusif, les blocs d'entrée et de sortie de la transformation non linéaire avec ledit nombre aléatoire.

2. Procédé selon la revendication 1, caractérisé en ce qu'il consiste à appliquer plusieurs (n-1) tours de chiffrement ou déchiffrement avec une clé (Ki) différente à chaque tour.

3. Procédé selon la revendication 1 ou 2, caractérisé en ce qu'il consiste à combiner le code d'entrée (S_0, S_n) avec un deuxième nombre aléatoire (R) de même dimension que ce code.

4. Procédé selon l'une quelconque des revendications 1 à 3, caractérisé en ce que ladite transformation non linéaire (SUBBYTES, INVSUBBYTES) consiste à utiliser une table ($SBOX_{R1,R2}$) de substitution des blocs du code d'entrée, calculée avec un troisième nombre aléatoire (R2) de même longueur que ledit code et dont tous les blocs ont la même valeur, ladite table ($SBOX_{R1,R2}$) respectant le fait que la transformation d'un code d'entrée, préalablement combiné par un OU-Exclusif avec le premier nombre aléatoire (R1), correspond au résultat de la combinaison par un OU-Exclusif de ce code d'entrée avec ledit troisième nombre aléatoire.

5. Procédé selon l'une quelconque des revendications 1 à 4, caractérisé en ce qu'il est appliqué à un algorithme de chiffrement de type AES.

6. Procédé selon la revendication 1, caractérisé en ce que ledit premier nombre aléatoire (R1) est changé à chaque tour de chiffrement.

7. Procédé selon la revendication 3, caractérisé en ce que ledit deuxième nombre aléatoire (R) est changé à chaque chiffrement d'une nouvelle donnée.

8. Procédé selon la revendication 4, caractérisé en ce que ledit troisième nombre aléatoire (R2) est changé à chaque tour de chiffrement.

9. Circuit intégré comprenant un bloc de chiffrement/ déchiffrement d'une donnée d'entrée (S_0, S_n) découpée en blocs de mêmes dimensions, caractérisé en ce qu'il comporte

des moyens de génération d'au moins un premier nombre aléatoire (R1) de même taille que la taille des blocs de la donnée d'entrée ; et

des moyens de combinaison dudit nombre aléatoire avec chaque bloc, en entrée et en sortie d'une transformation non linéaire (34, 54) mise en oeuvre par le chiffrement/ déchiffrement.

10. Circuit selon la revendication 8, caractérisé en ce qu'il comprend des moyens pour la mise en oeuvre du procédé selon l'une quelconque des revendications 1 à 7.

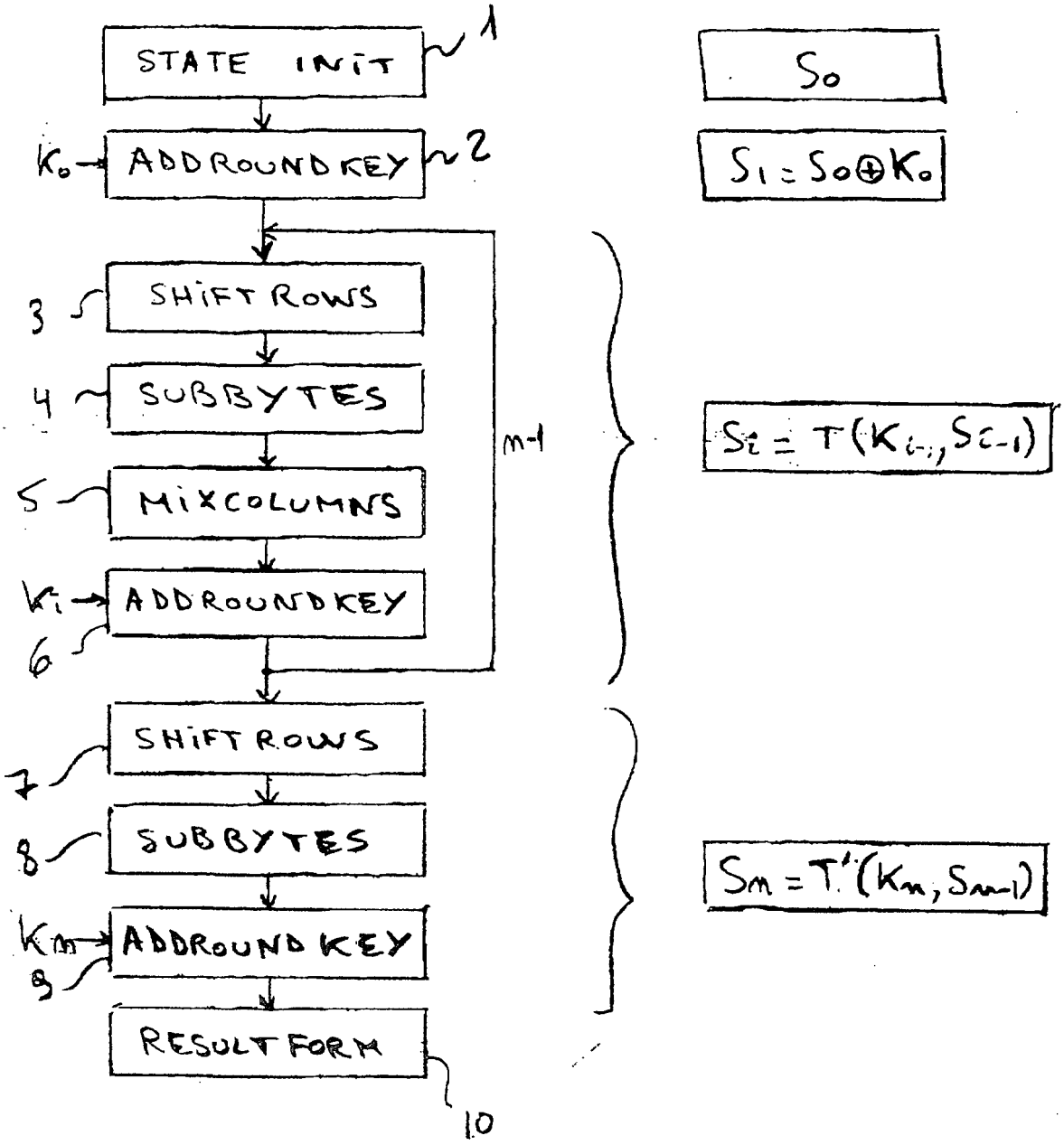

STATE INIT
1
S_0
K_0
ADDROUNDKEY
2
$S_1 = S_0 \oplus K_0$
SHIFT ROWS
3
SUBBYTES
4
MIXCOLUMNS
5
K_i
ADDROUNDKEY
6
m-1
$S_i = T(K_{i-1}, S_{i-1})$
SHIFT ROWS
7
SUBBYTES
8
K_m
ADDROUNDKEY
9
$S_m = T'(K_m, S_{m-1})$
RESULT FORM
10

FIG. 1

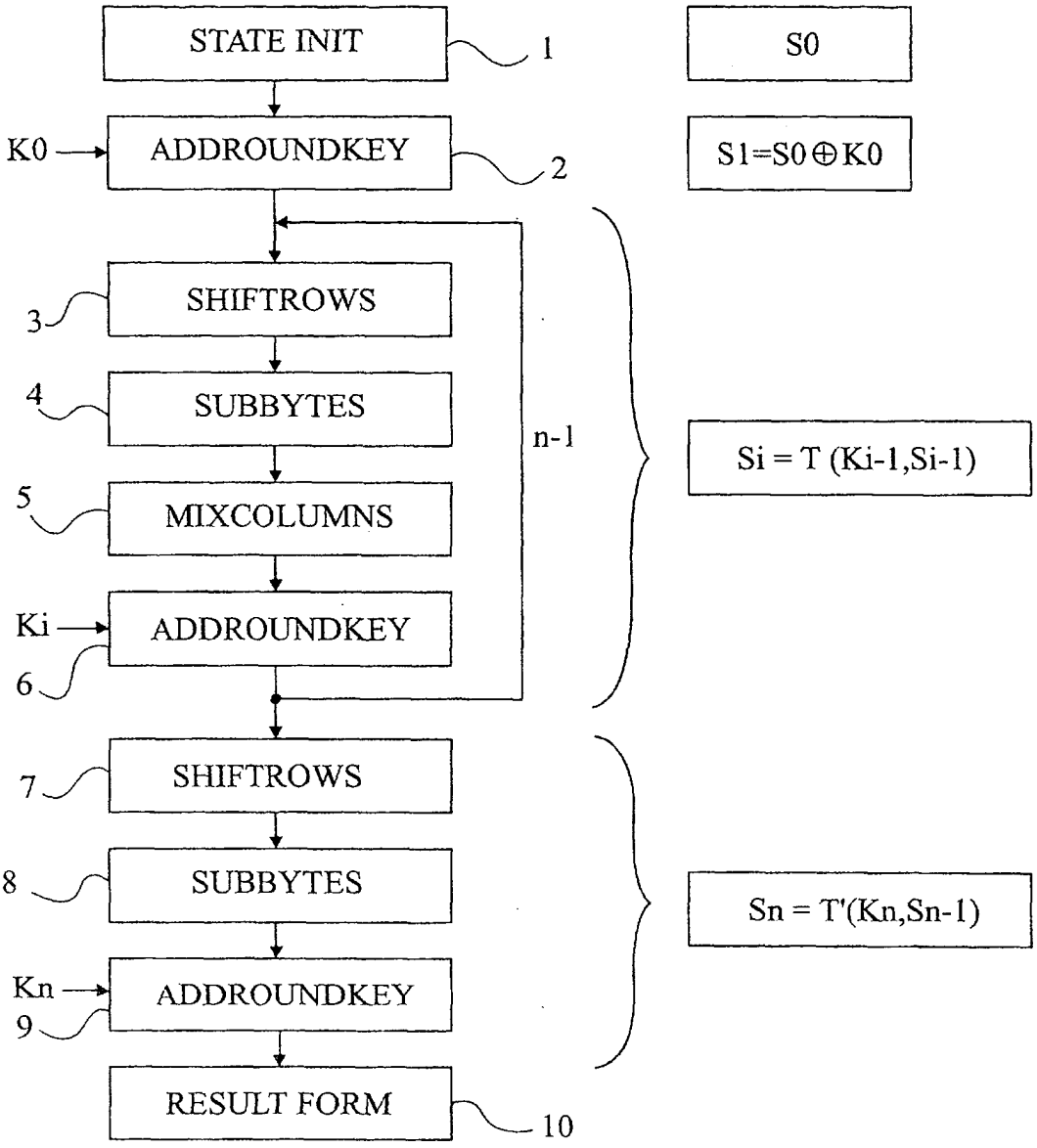
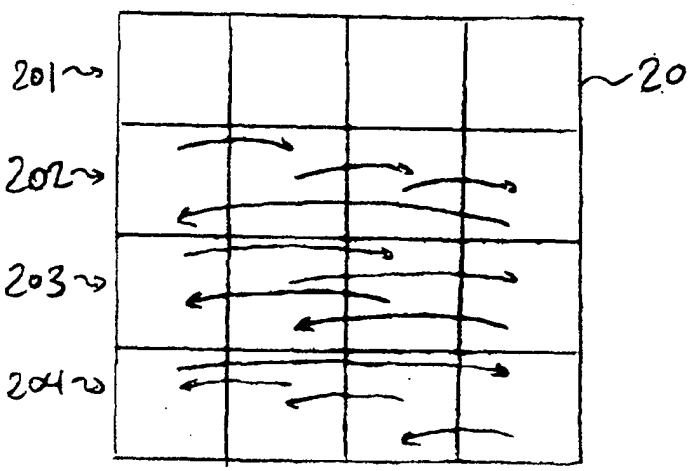
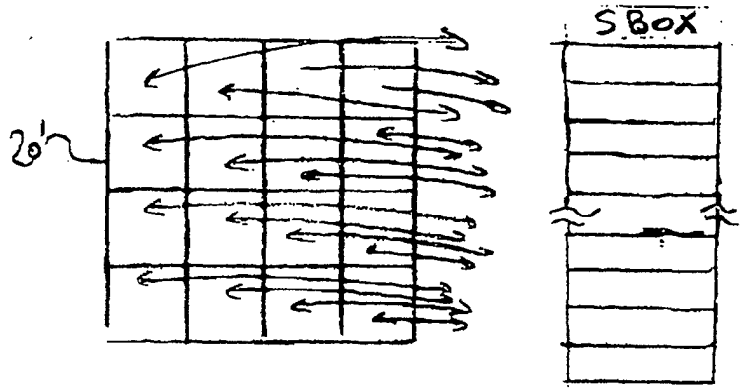
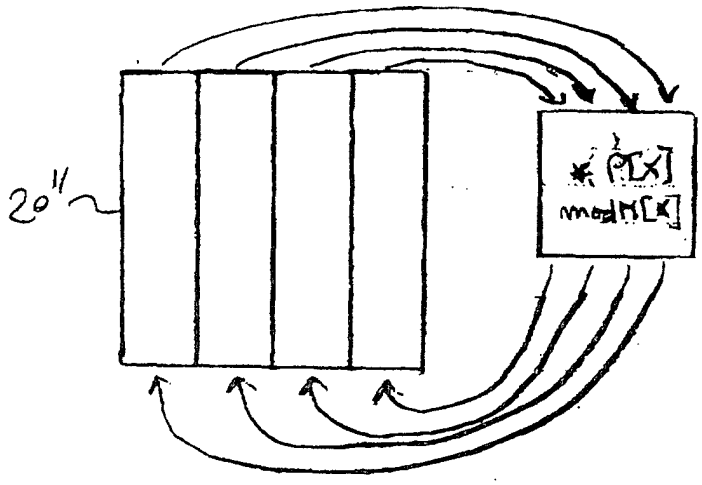
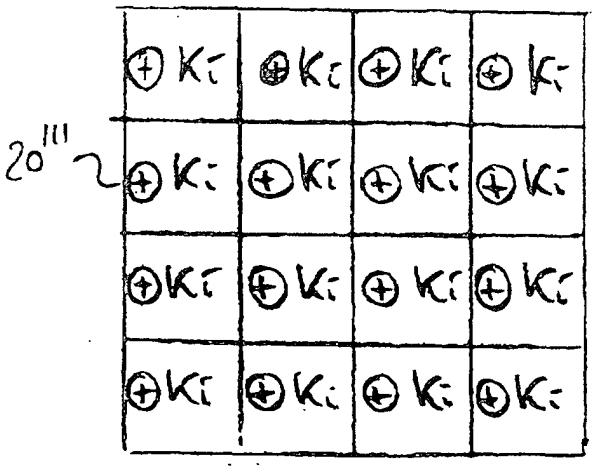

STATE INIT
1
S0
K0
ADDROUNDKEY
2
S1=S0⊕K0
SHIFTROWS
3
SUBBYTES
4
MIXCOLUMNS
5
Ki
ADDROUNDKEY
6
n-1
Si = T (Ki-1,Si-1)
SHIFTROWS
7
SUBBYTES
8
Kn
ADDROUNDKEY
9
Sn = T'(Kn,Sn-1)
RESULT FORM
10

Fig 1


201
202
203
204
20


20'
SBox


20''
x[PLX]
mod M[K]


20'''
⊕Ki

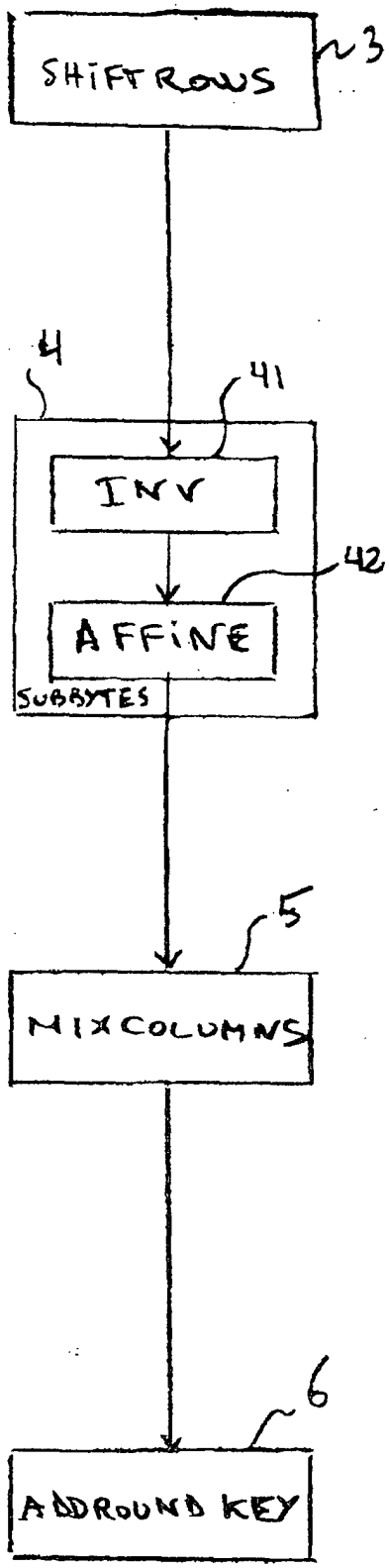

SHIFTROWS
3
4
41
INV
42
AFFINE
SUBBYTES
5
MIXCOLUMNS
6
ADDROUNDKEY

FIG. 2

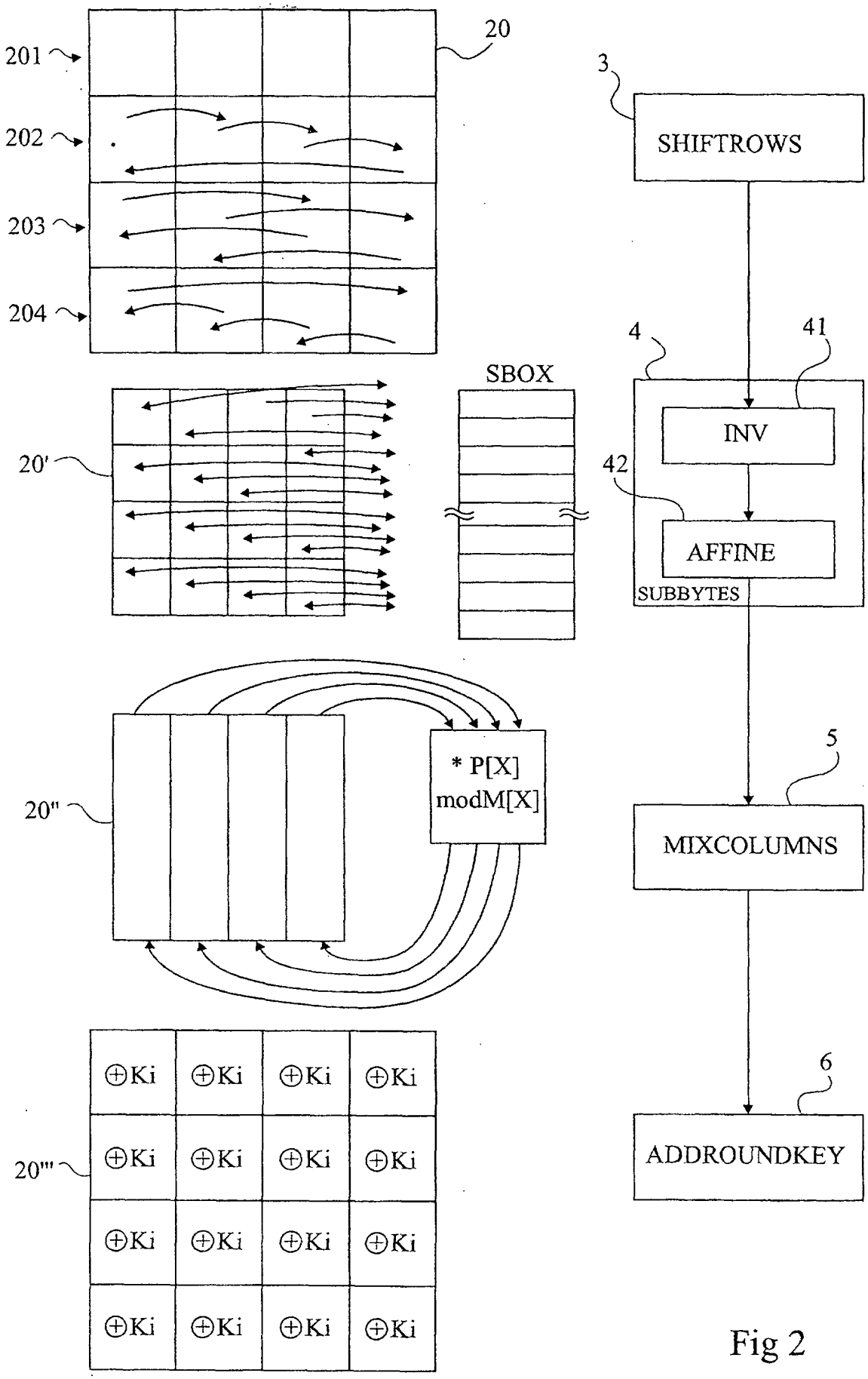

20
201
202
203
204
20'
SBOX
20"
* P[X]
modM[X]
20'''
⊕Ki
3
SHIFTROWS
4
41
INV
42
AFFINE
SUBBYTES
5
MIXCOLUMNS
6
ADDROUNDKEY

Fig 2

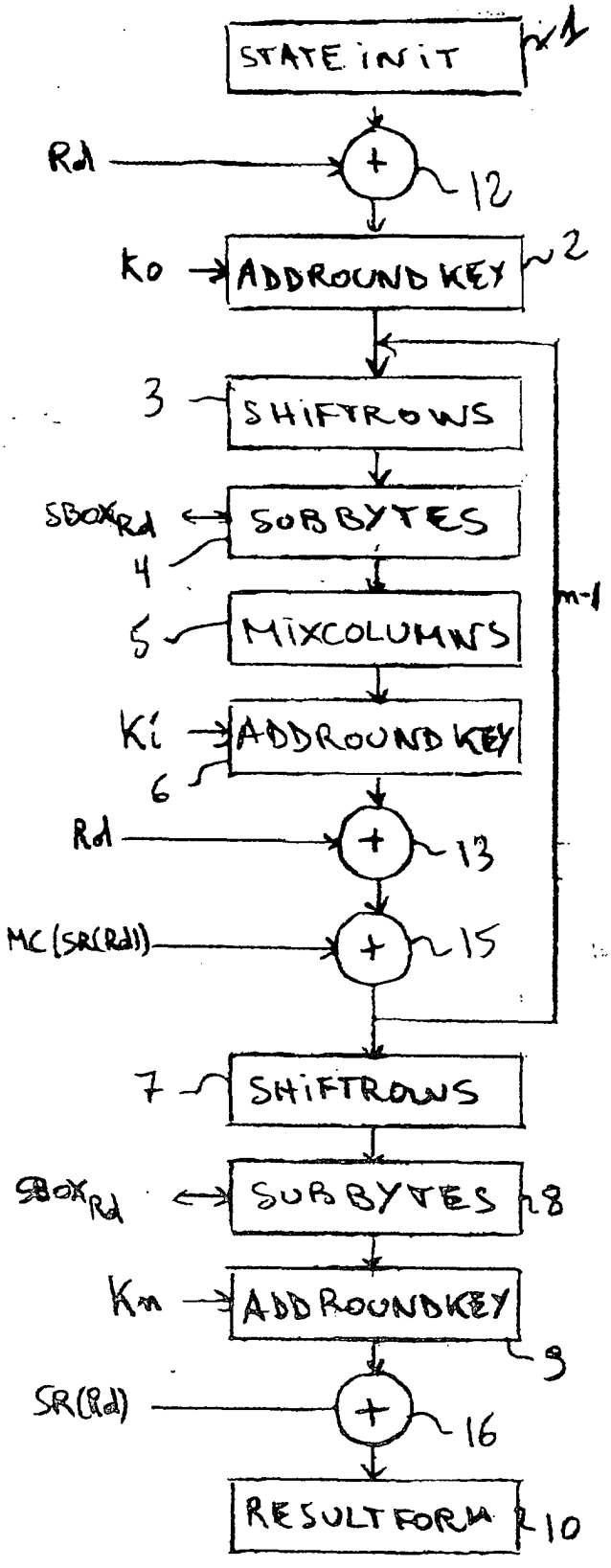

STATE INIT
1
Rd
+
12
K0
ADDROUNDKEY
2
3
SHIFTROWS
SBOX Rd
SUBBYTES
4
5
MIXCOLUMNS
n-1
Ki
ADDROUNDKEY
6
Rd
+
13
MC(SR(Rd))
+
15
7
SHIFTROWS
SBOX Rd
SUBBYTES
8
Km
ADDROUNDKEY
9
SR(Rd)
+
16
RESULTFORM
10

FIG-3

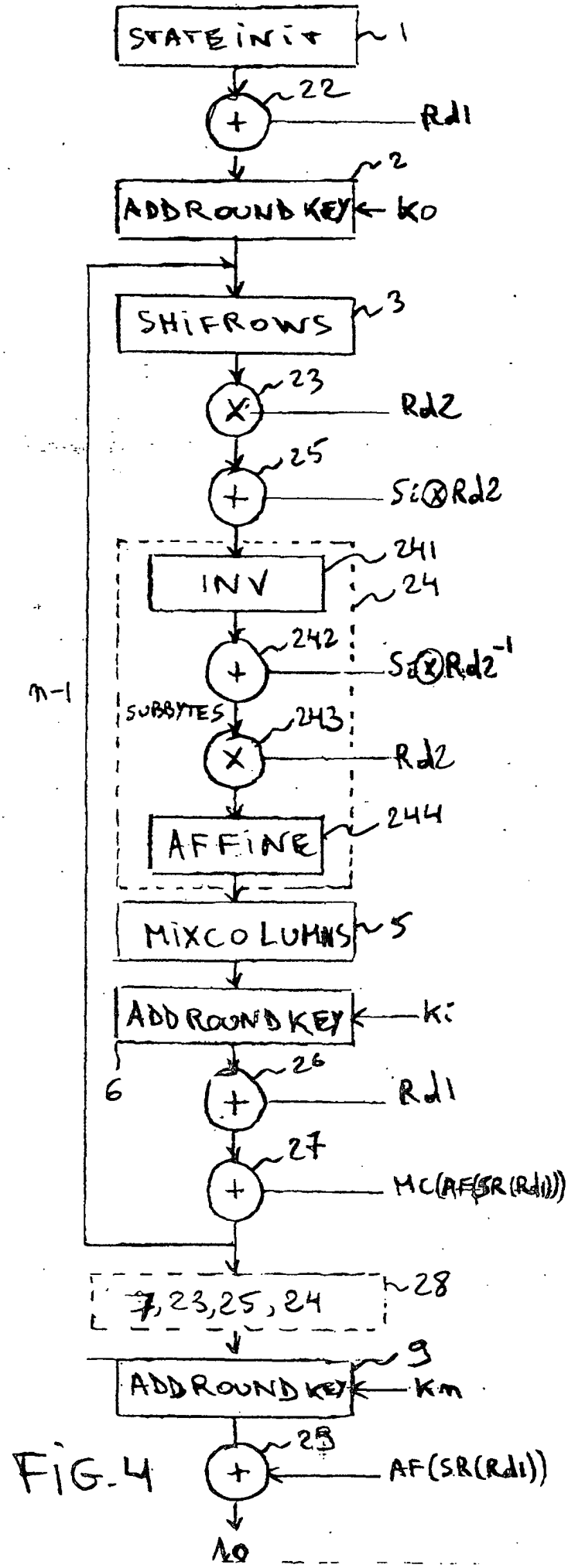

STATE INIT
1
22
+
Rd1
ADDROUNDKEY
2
K0
SHIFROWS
3
23
X
Rd2
25
+
Si⊗Rd2
241
INV
24
242
+
Si⊗Rd2^-1
n-1
SUBBYTES
243
X
Rd2
AFFINE
244
MIXCOLUMNS
5
ADDROUNDKEY
Ki
6
26
+
Rd1
27
+
MC(AF(SR(Rd1)))
7,23,25,24
28
ADDROUNDKEY
9
Km
29
+
AF(SR(Rd1))
10

FIG-4

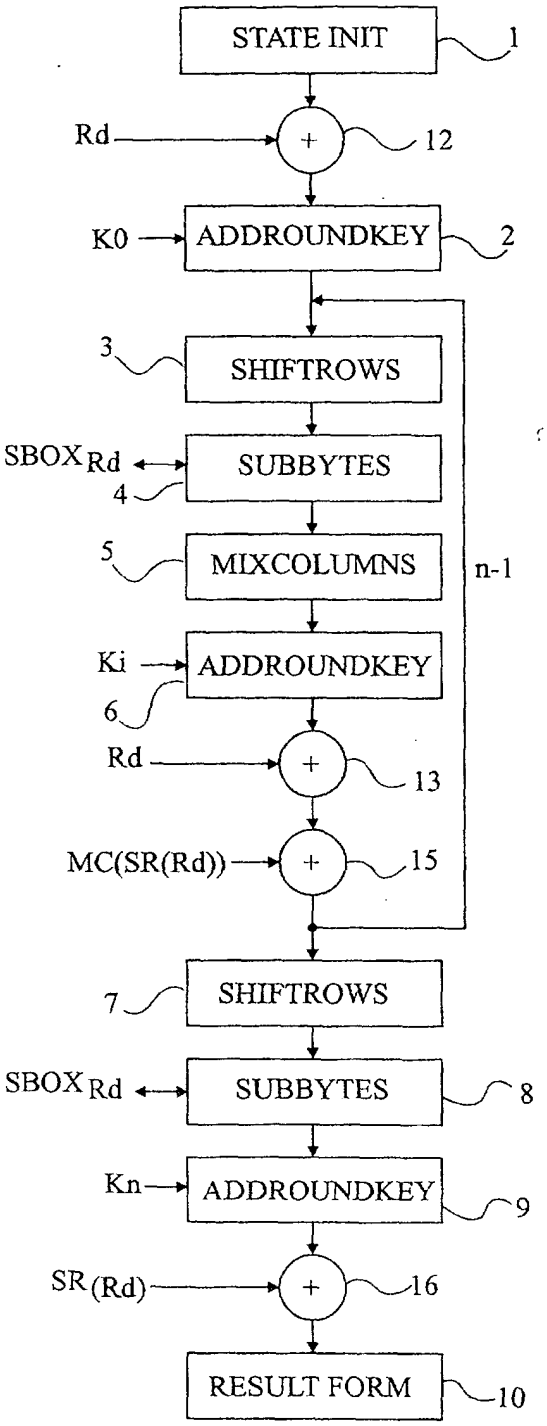

STATE INIT
1
Rd
+
12
K0
ADDROUNDKEY
2
3
SHIFTROWS
SBOX Rd
SUBBYTES
4
5
MIXCOLUMNS
n-1
Ki
ADDROUNDKEY
6
Rd
+
13
MC(SR(Rd))
+
15
7
SHIFTROWS
SBOX Rd
SUBBYTES
8
Kn
ADDROUNDKEY
9
SR(Rd)
+
16
RESULT FORM
10

Fig 3

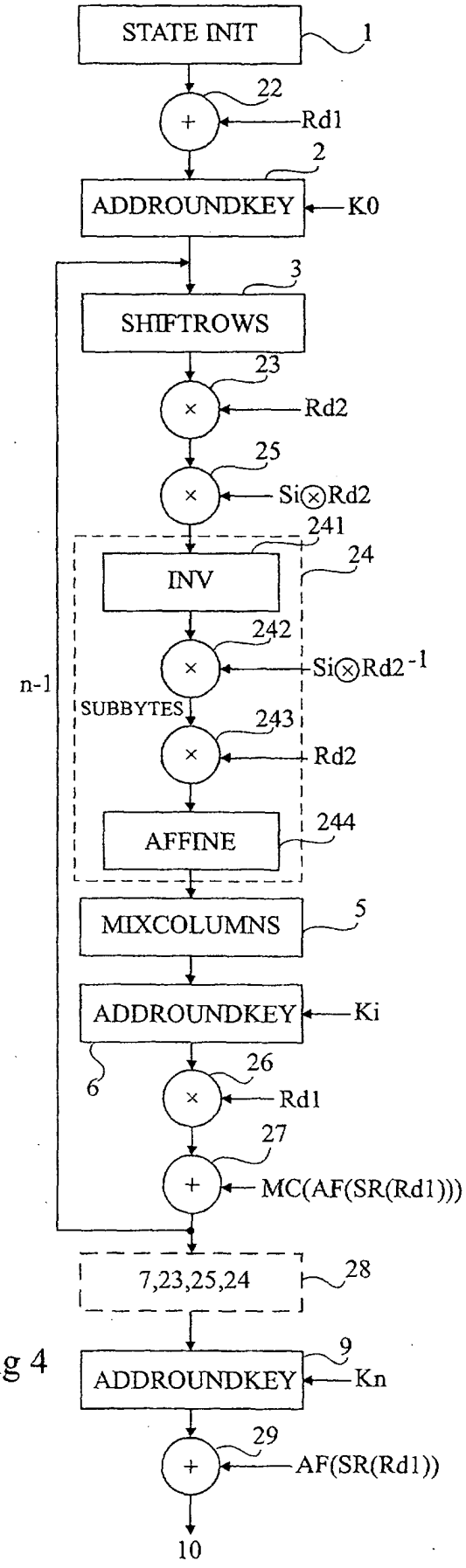

STATE INIT
1
22
+
Rd1
2
ADDROUNDKEY
K0
3
SHIFTROWS
23
×
Rd2
25
×
Si⊗Rd2
241
24
INV
242
×
Si⊗Rd2-1
n-1
SUBBYTES
243
×
Rd2
AFFINE
244
MIXCOLUMNS
5
ADDROUNDKEY
Ki
6
26
×
Rd1
27
+
MC(AF(SR(Rd1)))
7,23,25,24
28
9
ADDROUNDKEY
Kn
29
+
AF(SR(Rd1))
10

Fig 4

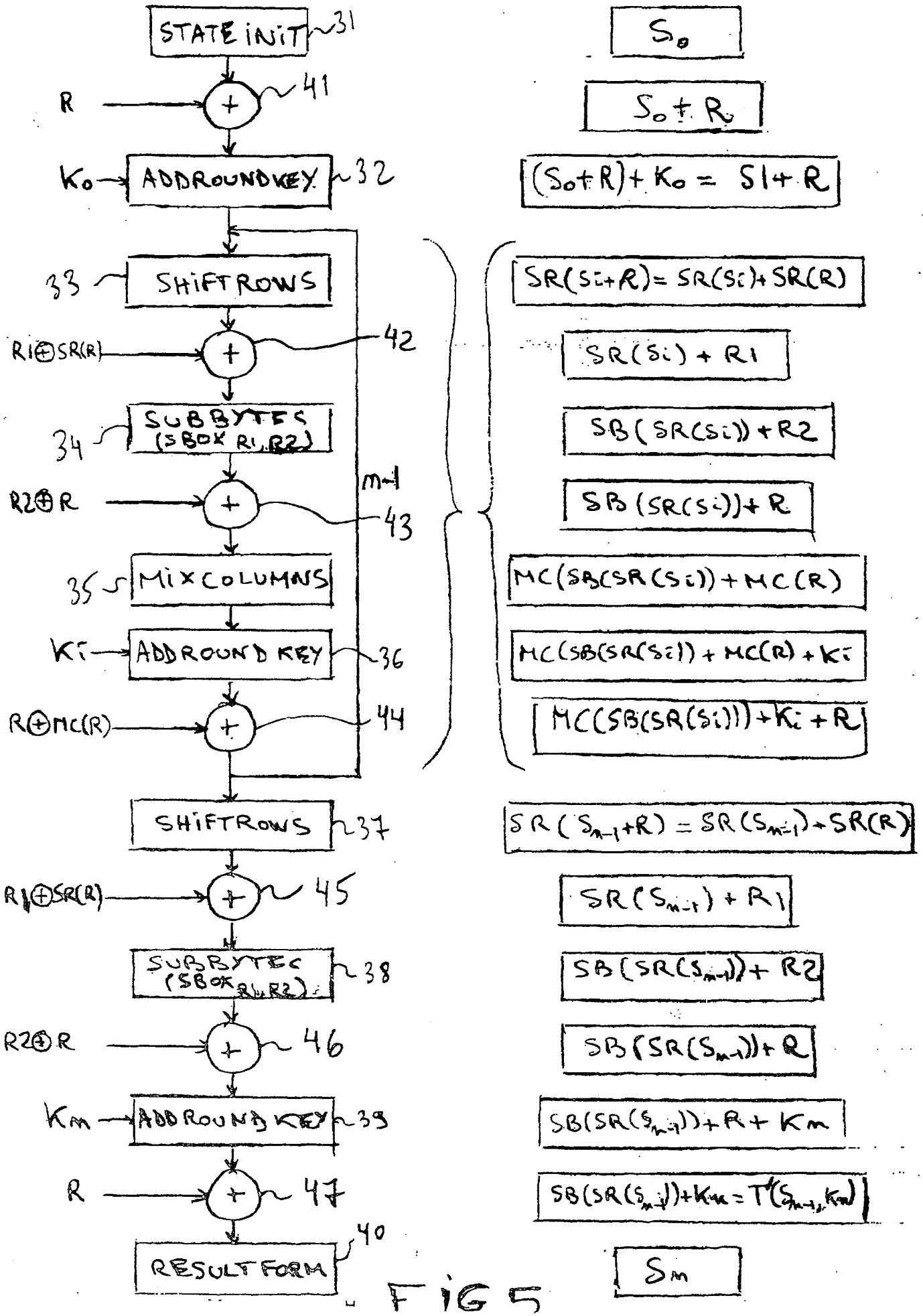

STATE INIT
31
S_0
R
41
$S_0 + R$
K_0
ADDROUNDKEY
32
$(S_0+R)+K_0 = S1+R$
33
SHIFTROWS
$SR(S_i+R) = SR(S_i)+SR(R)$
R1⊕SR(R)
42
$SR(S_i) + R1$
34
SUBBYTES (SBOX R1,R2)
$SB(SR(S_i)) + R2$
m-1
R2⊕R
43
$SB(SR(S_i)) + R$
35
MIXCOLUMNS
$MC(SB(SR(S_i)) + MC(R)$
K_i
ADDROUNDKEY
36
$MC(SB(SR(S_i)) + MC(R) + K_i$
R⊕MC(R)
44
$MC(SB(SR(S_i))) + K_i + R$
SHIFTROWS
37
$SR(S_{m-1}+R) = SR(S_{m-1}) + SR(R)$
R1⊕SR(R)
45
$SR(S_{m-1}) + R1$
SUBBYTES (SBOX R1,R2)
38
$SB(SR(S_{m-1})) + R2$
R2⊕R
46
$SB(SR(S_{m-1})) + R$
K_m
ADDROUNDKEY
39
$SB(SR(S_{m-1})) + R + K_m$
R
47
$SB(SR(S_{m-1})) + K_m = T(S_{m-1}, K_m)$
RESULTFORM
40
S_m

FIG 5

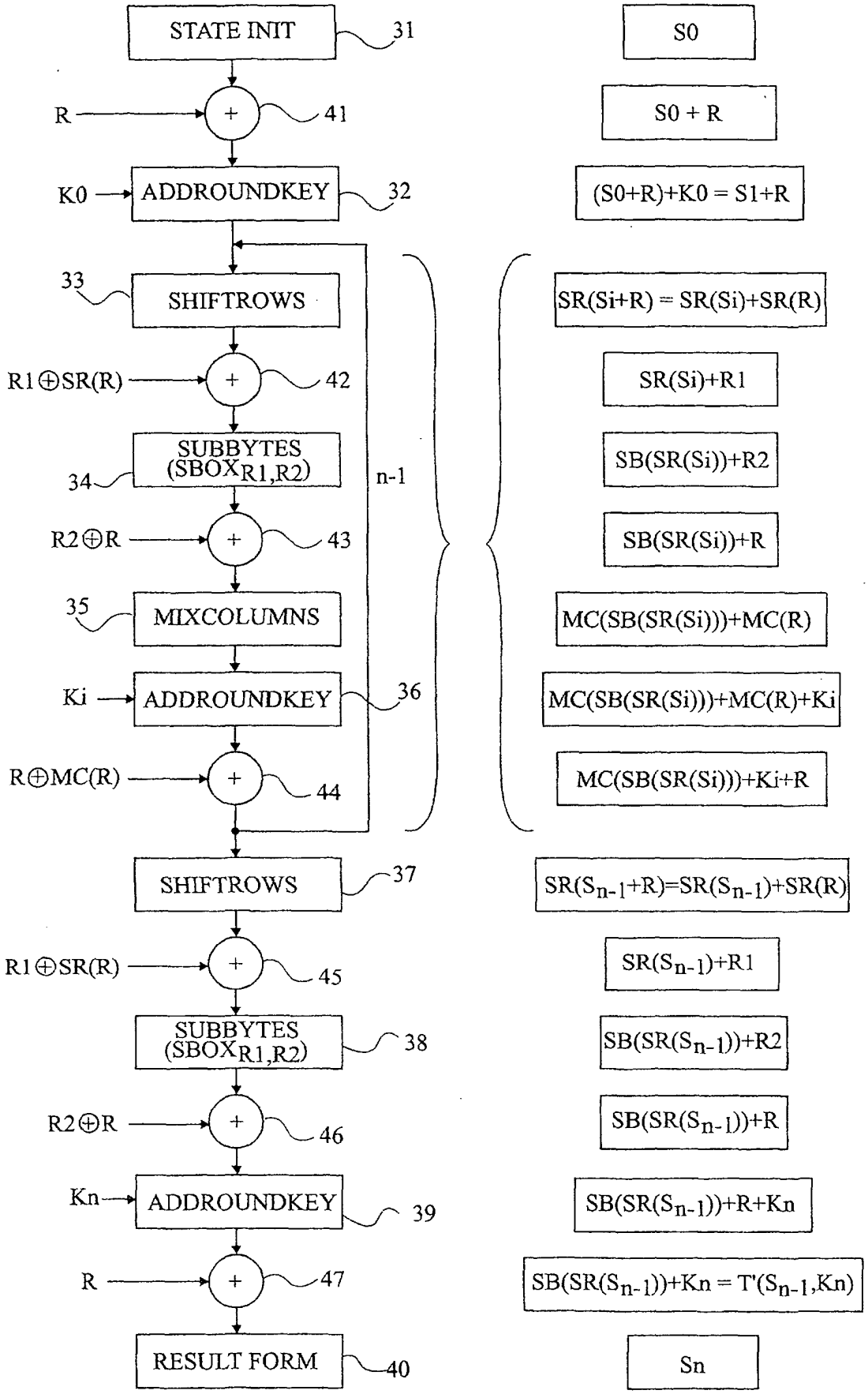

STATE INIT
31
S0
R
+
41
S0 + R
K0
ADDROUNDKEY
32
(S0+R)+K0 = S1+R
33
SHIFTROWS
SR(Si+R) = SR(Si)+SR(R)
R1⊕SR(R)
+
42
SR(Si)+R1
SUBBYTES (SBOXR1,R2)
34
SB(SR(Si))+R2
n-1
R2⊕R
+
43
SB(SR(Si))+R
35
MIXCOLUMNS
MC(SB(SR(Si)))+MC(R)
Ki
ADDROUNDKEY
36
MC(SB(SR(Si)))+MC(R)+Ki
R⊕MC(R)
+
44
MC(SB(SR(Si)))+Ki+R
SHIFTROWS
37
SR(Sn-1+R)=SR(Sn-1)+SR(R)
R1⊕SR(R)
+
45
SR(Sn-1)+R1
SUBBYTES (SBOXR1,R2)
38
SB(SR(Sn-1))+R2
R2⊕R
+
46
SB(SR(Sn-1))+R
Kn
ADDROUNDKEY
39
SB(SR(Sn-1))+R+Kn
R
+
47
SB(SR(Sn-1))+Kn = T'(Sn-1,Kn)
RESULT FORM
40
Sn

Fig 5

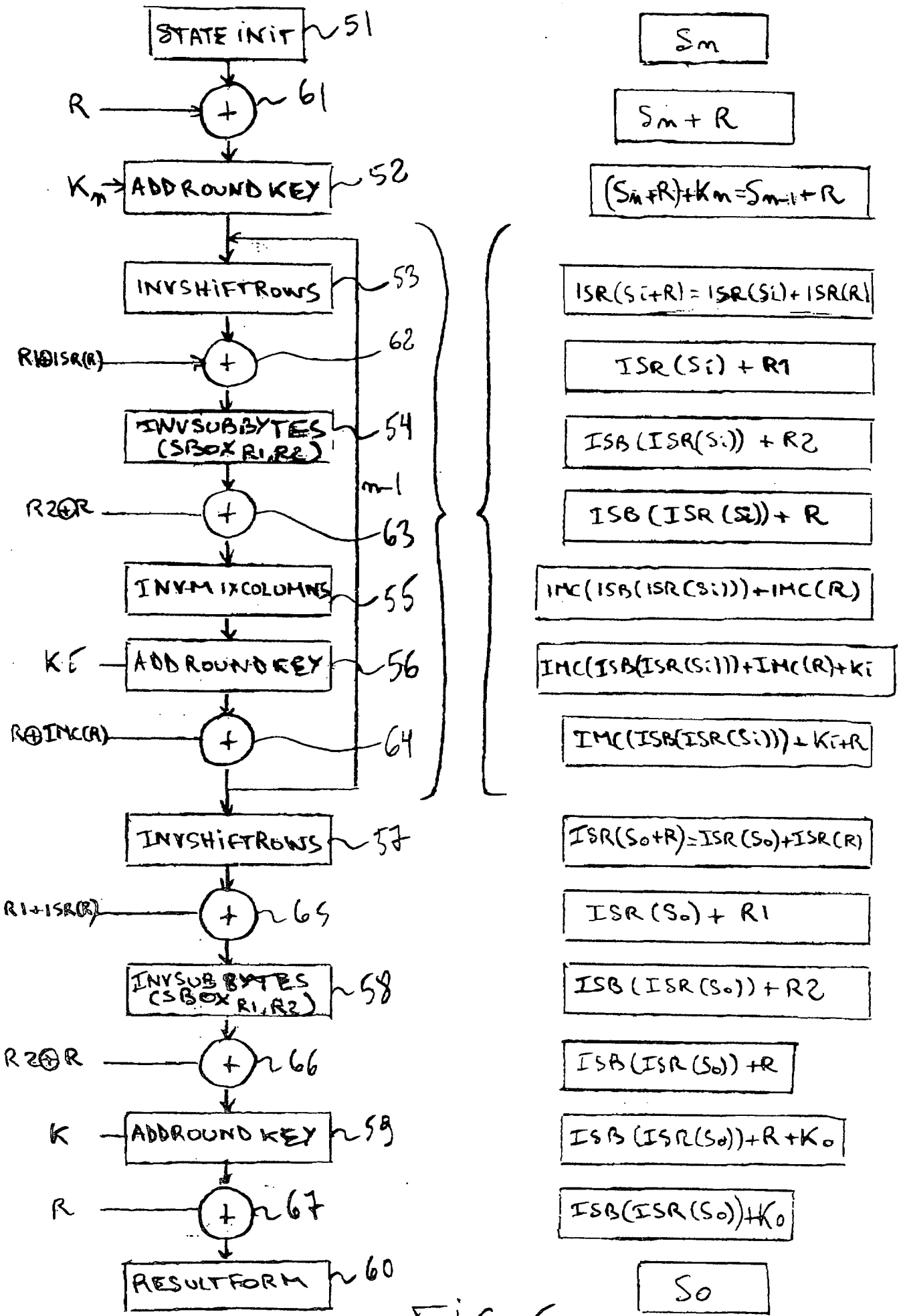

STATE INIT
51
Sm
R
61
Sm + R
Km
ADD ROUND KEY
52
(Sm+R)+Km=Sm-1+R
INVSHIFTROWS
53
ISR(Si+R) = ISR(Si)+ISR(R)
R1⊕ISR(R)
62
ISR(Si) + R1
INVSUBBYTES (SBOX R1,R2)
54
ISB(ISR(Si)) + R2
m-1
R2⊕R
63
ISB(ISR(Si)) + R
INVMIXCOLUMNS
55
IMC(ISB(ISR(Si)))+IMC(R)
Ki
ADD ROUND KEY
56
IMC(ISB(ISR(Si)))+IMC(R)+Ki
R⊕IMC(R)
64
IMC(ISB(ISR(Si)))+Ki+R
INVSHIFTROWS
57
ISR(So+R)=ISR(So)+ISR(R)
R1+ISR(R)
65
ISR(So) + R1
INVSUBBYTES (SBOX R1,R2)
58
ISB(ISR(So)) + R2
R2⊕R
66
ISB(ISR(So)) +R
K
ADDROUNDKEY
59
ISB(ISR(So))+R+Ko
R
67
ISB(ISR(So))+Ko
RESULTFORM
60
So

FIG. 6

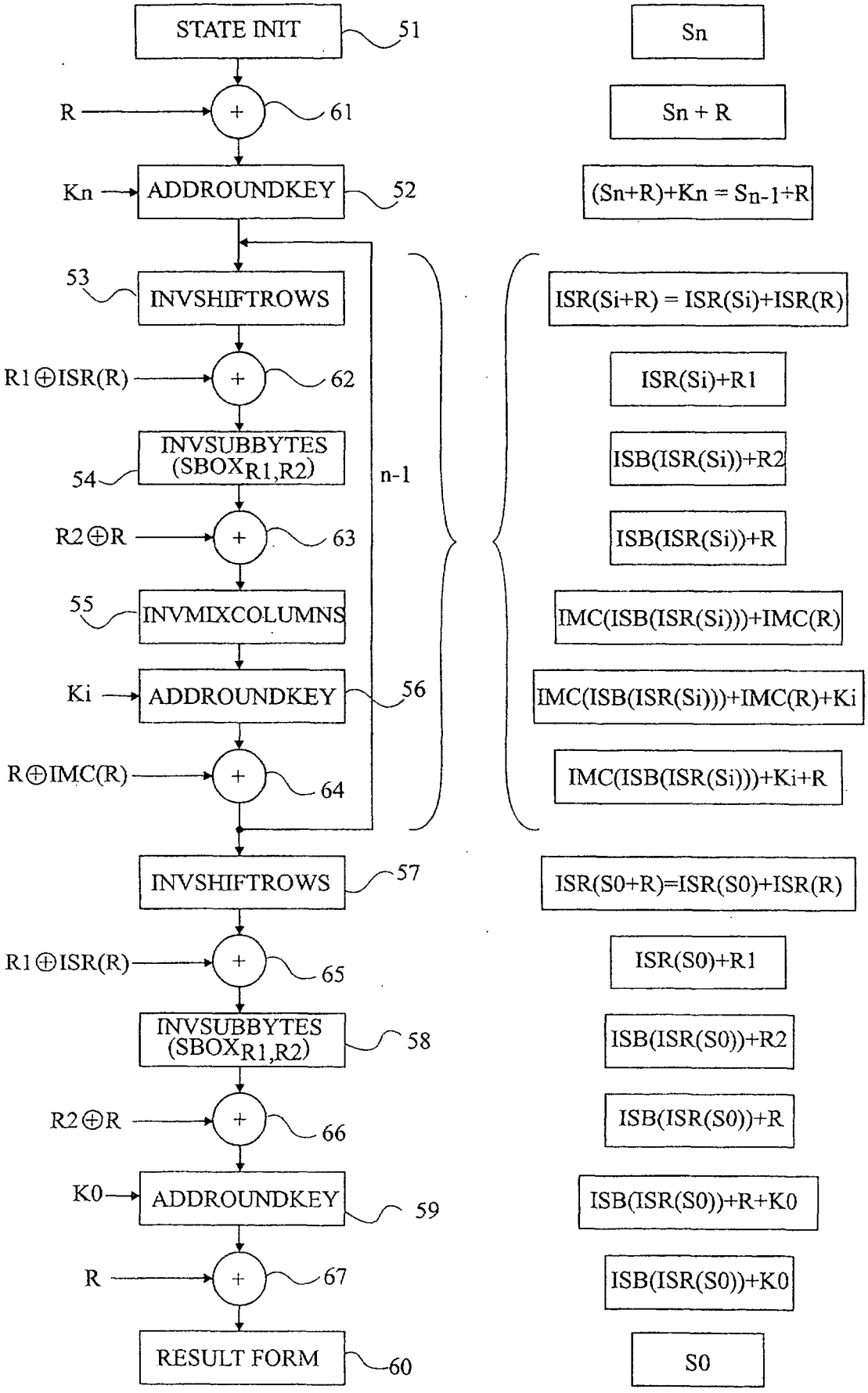

STATE INIT
51
Sn
R
+
61
Sn + R
Kn
ADDROUNDKEY
52
(Sn+R)+Kn = Sn-1+R
53
INVSHIFTROWS
ISR(Si+R) = ISR(Si)+ISR(R)
R1⊕ISR(R)
+
62
ISR(Si)+R1
INVSUBBYTES
(SBOXR1,R2)
54
ISB(ISR(Si))+R2
n-1
R2⊕R
+
63
ISB(ISR(Si))+R
55
INVMIXCOLUMNS
IMC(ISB(ISR(Si)))+IMC(R)
Ki
ADDROUNDKEY
56
IMC(ISB(ISR(Si)))+IMC(R)+Ki
R⊕IMC(R)
+
64
IMC(ISB(ISR(Si)))+Ki+R
INVSHIFTROWS
57
ISR(S0+R)=ISR(S0)+ISR(R)
R1⊕ISR(R)
+
65
ISR(S0)+R1
INVSUBBYTES
(SBOXR1,R2)
58
ISB(ISR(S0))+R2
R2⊕R
+
66
ISB(ISR(S0))+R
K0
ADDROUNDKEY
59
ISB(ISR(S0))+R+K0
R
+
67
ISB(ISR(S0))+K0
RESULT FORM
60
S0

Fig 6

INPI
INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE

DÉPARTEMENT DES BREVETS
26 bis, rue de Saint Pétersbourg
75800 Paris Cedex 08
Téléphone : 01 53 04 53 04 Télécopie : 01 42 94 86 54

BREVET D'INVENTION, CERTIFICAT D'UTILITÉ

Code de la propriété intellectuelle-Livre VI

N° 55 -1328

DÉSIGNATION D'INVENTEUR(S) PAGE N°1/ 2

(Si le demandeur n'est pas l'inventeur ou l'unique inventeur)

Cet imprimé est à remplir lisiblement à l'encre noire

Vos références pour ce dossier *(facultatif)*	B5532
N° D'ENREGISTREMENT NATIONAL	0208268

TITRE DE L'INVENTION (200 caractères ou espaces maximum)

CHIFFREMENT/DÉCHIFFREMENT EXÉCUTÉ PAR UN CIRCUIT INTÉGRÉ

LE(S) DEMANDEUR(S) :

STMicroelectronics SA

DESIGNE (NT) EN TANT QU'INVENTEUR(S) : (Indiquez en haut à droite "Page N°1/1" S'il y a plus de trois inventeurs, utilisez un formulaire identique et numérotez chaque page en indiquant le nombre total de pages).

Prénoms & Nom		Pierre-Yvan Liardet	
ADRESSE	Rue	56, Rue du Pralou, Lotissement L'Audiguier	
	Code postal et ville	13790	PEYNIER, FRANCE
Société d'appartenance *(facultatif)*			
Prénoms & Nom		Fabrice Romain	
ADRESSE	Rue	Les Héliades Bât. A, 535, Avenue de Bagatelle	
	Code postal et ville	13090	AIX EN PROVENCE, FRANCE
Société d'appartenance *(facultatif)*			
Prénoms & Nom		Yannick Teglia	
ADRESSE	Rue	22, Traverse de la Dominique, Bâtiment B	
	Code postal et ville	13011	MARSEILLE, FRANCE
Société d'appartenance (facultatif)			

**DATE ET SIGNATURE (S)
DU (DES) DEMANDEUR(S)
OU DU MANDATAIRE
(Nom et qualité du signataire)**

**Michel de Beaumont
Mandataire n° 92-1016**

Le 2 juillet 2002

La loi n°78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés s'applique aux réponses faites à ce formulaire. Elle garantit un droit d'accès et de rectification pour les données vous concernant auprès de l'INPI.

INPI
INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE
DÉPARTEMENT DES BREVETS
26 bis, rue de Saint Pétersbourg
75800 Paris Cedex 08
Téléphone : 01 53 04 53 04 Télécopie : 01 42 94 86 54

BREVET D'INVENTION, CERTIFICAT D'UTILITÉ

Code de la propriété intellectuelle-Livre VI

cerfa N° 55 -1328

DÉSIGNATION D'INVENTEUR(S) PAGE N°2/2

(Si le demandeur n'est pas l'inventeur ou l'unique inventeur)

Cet imprimé est à remplir lisiblement à l'encre noire

Vos références pour ce dossier *(facultatif)*	B5532
N° D'ENREGISTREMENT NATIONAL	0208268

TITRE DE L'INVENTION (200 caractères ou espaces maximum)

CHIFFREMENT/DÉCHIFFREMENT EXÉCUTÉ PAR UN CIRCUIT INTÉGRÉ

LE(S) DEMANDEUR(S) :

STMicroelectronics SA

DESIGNE (NT) EN TANT QU'INVENTEUR(S) : (Indiquez en haut à droite "Page N°1/1" S'il y a plus de trois inventeurs, utilisez un formulaire identique et numérotez chaque page en indiquant le nombre total de pages).

Prénoms & Nom		Laurence Sirtori	
ADRESSE	Rue	641, Chemin de Grisole	
	Code postal et ville	13530	TRETS, FRANCE
Société d'appartenance *(facultatif)*			
Prénoms & Nom			
ADRESSE	Rue		
	Code postal et ville		
Société d'appartenance *(facultatif)*			
Prénoms & Nom			
ADRESSE	Rue		
	Code postal et ville		
Société d'appartenance (facultatif)			

DATE ET SIGNATURE (S)
DU (DES) DEMANDEUR(S)
OU DU MANDATAIRE
(Nom et qualité du signataire)

Michel de Beaumont
Mandataire n° 92-1016

Le 2 juillet 2002